

09/730,190

MS160309.01/MSFTP170US

AMENDMENTS TO THE CLAIMS

This listing of claims will replace all prior versions, and listings, of claims in the application:

Listing of Claims:

1. (Currently amended): A client side HTTP stack software component effectuated on a machine that processes ~~for processing~~ requests, comprising:
 - at least one completion port object;
 - a thread pool comprising a plurality of threads that adapted to ~~adapted to~~ process tasks associated with at least one client side request; and
 - a client side state machine associated with the at least one request.
2. (Currently amended): The client side HTTP stack implementation of claim 1, further comprising a scheduler thread ~~adapted to~~ that activates an object scheduled to begin sending requests at a specific time.
3. (Currently amended): The client side HTTP stack implementation of claim 1, further comprising a DNS thread ~~adapted to~~ that resolves domain names into IP addresses.
4. (Currently amended): The client side HTTP stack implementation of claim 1, further comprising a timeout thread with a list of active sockets and timers associated with each socket, ~~and adapted to~~ the timeout thread selectively times-out ~~timeout~~ at least one socket according to at least one timer in the list.
5. (Currently amended): The client side HTTP stack implementation of claim 4, further comprising a scheduler thread ~~adapted to~~ that activates an object scheduled to begin sending requests at a specific time.

09/730,190

MS160309.01/MSFTP170US

6. (Currently amended): The client side HTTP stack implementation of claim 5, further comprising a DNS thread ~~adapted to~~ that resolves domain names into IP addresses.

7. (Currently amended): The client side HTTP stack implementation of claim 4, further comprising a DNS thread ~~adapted to~~ that resolves domain names into IP addresses.

8. (Currently amended): A machine effectuated software component that implements ~~for implementing~~ a client side HTTP stack, comprising:
a thread pool comprising N threads that ~~adapted to~~ process M requests from a client application component, ~~wherein~~ where N and M are integers greater than 1 and ~~wherein~~ where M is greater than N; and
a state machine associated with each of the M requests.

9. (Currently amended): The software component of claim 8, further comprising at least one thread activation component ~~adapted to~~ that activates at least one of the N threads based on an event.

10. (Currently amended): The software component of claim 9, ~~wherein~~ the at least one thread activation component is a completion port.

11. (Currently amended): The software component of claim 9, ~~wherein~~ at least one of the N threads is ~~adapted to~~ deactivates itself and returns to the thread pool when an operation being processed by the at least one of the threads is pending.

12. (Currently amended): The software component of claim 11, ~~wherein~~ the event is the receipt of a completion packet by the at least one thread activation component.

09/730,190

MS160309.01/MSFTP170US

13. (Currently amended): The software component of claim 12, ~~wherein~~ the at least one thread activation component is a completion port.

14. (Currently amended): The software component of claim 13, further comprising a scheduler thread ~~adapted to that~~ activates an object scheduled to begin sending requests at a specific time.

15. (Currently amended): The software component of claim 14, further comprising a DNS thread ~~adapted to that~~ resolves domain names into IP addresses.

16. (Currently amended): The software component of claim 15, further comprising a timeout thread with a list of active sockets and timers associated with each socket, ~~and adapted to the timeout thread~~ selectively ~~timeout~~ times-out at least one socket according to at least one timer in the list.

17. (Cancelled).

18. (Currently amended): The software component of claim 9 [[17]], further comprising at least one key associated with [[the]] at least one of the M requests, wherein a first one of the N threads is associated with the at least one of the M requests, and ~~wherein~~ the thread activation component ~~is adapted to~~ associates the context of the first one of the N threads with the at least one state machine using the at least one key, in order to activate the first one of the N threads.

19. (Currently amended): The software component of claim 18, ~~wherein~~ the thread activation component ~~is adapted to~~ associates the context of one of the N threads with the at least one state machine using the at least one key in order to activate the one of the N threads based on an event.

09/730,190

MS160309.01/MSFTP170US

20. (Currently amended): The software component of claim 8, further comprising a scheduler thread ~~adapted to~~ that ~~activates~~ an object scheduled to begin sending requests at a specific time.

21. (Currently amended): The software component of claim 8, further comprising a DNS thread ~~adapted to~~ that resolves domain names into IP addresses.

22. (Currently amended): The software component of claim 8, further comprising a timeout thread with a list of active sockets and timers associated with each socket, ~~and adapted to~~ the timeout thread ~~selectively timeout~~ times-out at least one socket according to at least one timer in the list.

23. (Currently amended): A method effectuated at least in part by a machine for ~~for~~ [[of]] implementing a client side HTTP stack, comprising:

processing M requests from a client application component using a thread pool comprising N threads, ~~wherein~~ where M and N are integers greater than 1 and ~~wherein~~ where M is greater than N; and

associating a state machine with each of the M requests.

24. (Original): The method of claim 23, further comprising:
selectively deactivating at least one of the N threads; and
activating at least another of the N threads based on an event using at least one thread activation component.

25. (Currently amended): The method of claim 24, ~~wherein~~ the at least one thread activation component is a completion port.

26. (Currently amended): The method of claim 24, ~~wherein~~ selectively deactivating at least one of the N threads comprises deactivating the at least one of the N threads when an operation being processed by the at least one of the N threads is pending.

09/730,190

MS160309.01/MSFTP170US

27. (Currently amended): The method of claim 26, ~~wherein~~ activating at least another of the N threads based on an event comprises:

receiving a completion packet using the thread activation component; and
activating one of the N threads upon receipt of the completion packet using the thread activation component.

28. (Currently amended): The method of claim 27, wherein the at least one thread activation component is a completion port.

29. (Original): The method of claim 28, further comprising activating an object scheduled to begin sending requests at a specific time using a scheduler thread.

30. (Original): The method of claim 29, further comprising resolving domain names into IP addresses using a DNS thread.

31. (Original): The method of claim 30, further comprising selectively timing out at least one socket according to at least one timer associated with the at least one socket using a timeout thread comprising a list of active sockets and timers associated with each socket.

32. (Cancelled).

33. (Currently amended): The method of claim 23 ~~[[32]]~~, further comprising:
associating at least one key with ~~[[the]]~~ at least one of the M requests;
associating a first one of the N threads with the at least one of the M requests; and
associating a context of the first one of the N threads with the at least one state machine using the at least one key, in order to deactivate the first one of the N threads.

34. (Original): The method of claim 33, further comprising associating a context of one of the N threads with the at least one state machine using the at least one key in order to activate the one of the N threads based on an event.

09/730,190

MS160309.01/MSFTP170US

35. (Currently amended): A computer-readable medium having computer-executable instructions for processing M requests from a client application component using a thread pool comprising N threads, ~~wherein~~ where M and N are integers greater than 1 and ~~wherein~~ where M is greater than N, and associating a state machine with at least one of the M requests.

36. (Original): The computer-readable medium of claim 35, further comprising computer-executable instructions for:
selectively deactivating at least one of the N threads; and
activating at least another of the N threads based on an event using at least one thread activation component.

37. (Currently amended): The computer-readable medium of claim 36, ~~wherein~~ the at least one thread activation component is a completion port.

38. (Currently amended): The computer-readable medium of claim 36, ~~wherein~~ the computer-executable instructions for selectively deactivating at least one of the N threads comprises computer-executable instructions for deactivating the at least one of the N threads when an operation being processed by the at least one of the N threads is pending.

39. (Currently amended): The computer-readable medium of claim 38, ~~wherein~~ the computer-executable instructions for activating at least another of the N threads based on an event comprises computer-executable instructions for:
receiving a completion packet using the thread activation component; and
activating one of the N threads upon receipt of the completion packet using the thread activation component.

09/730,190

MS160309.01/MSFTP170US

40. (Original): The computer-readable medium of claim 39, further comprising computer-executable instructions for activating an object scheduled to begin sending requests at a specific time using a scheduler thread.

41. (Original): The computer-readable medium of claim 40, further comprising computer-executable instructions for resolving domain names into IP addresses using a DNS thread.

42. (Original): The computer-readable medium of claim 41, further comprising computer-executable instructions for selectively timing out at least one socket according to at least one timer associated with the at least one socket using a timeout thread comprising a list of active sockets and timers associated with each socket.

43. (Cancelled).

44. (Currently amended): The computer-readable medium of claim 35 ~~[[43]]~~, further comprising computer-executable instructions for:
associating at least one key with the at least one of the M requests;
associating a first one of the N threads with the at least one of the M requests; and
associating a context of the first one of the N threads with the at least one state machine using the at least one key, in order to deactivate the first one of the N threads.

45. (Original): The computer-readable medium of claim 44, further comprising computer-executable instructions for associating a context of one of the N threads with the at least one state machine using the at least one key in order to activate the one of the N threads based on an event.

46. (Currently amended): A machine executed software component for implementing a client side HTTP stack, comprising:

09/730,190

MS160309.01/MSFTP170US

means for processing M requests from a client application component using a thread pool comprising N threads, ~~wherein~~ where M and N are integers greater than 1 and ~~wherein~~ where M is greater than N; and

means for assigning each of the M requests with a state machine.

47. (Original): The software component of claim 46, further comprising:
means for selectively deactivating at least one of the N threads; and
means for activating at least another of the N threads based on an event.

48. (Original): The software component of claim 47, further comprising
means for activating an object scheduled to begin sending requests at a specific time.

49. (Original): The software component of claim 47, further comprising
means for resolving domain names into IP addresses.

50. (Original): The software component of claim 47, further comprising
means for selectively timing out at least one socket according to at least one timer
associated with the at least one socket.